Capítulo 7

Sistema de reconocimiento de Voz

Como se mencionó en el capítulo anterior, dos fonemas, debido a los problemas de discapacidad pueden presentar la misma señal en frecuencia, por lo que podría en apariencia parecer imposible diferenciar uno del otro. No obstante el espectro en frecuencia de una señal de voz presenta máximos y mínimos distribuidos de diferente manera según sea el sonido analizado y el niño que produjo dicho sonido. Es decir, que el mismo fonema no presenta la misma representación si lo pronuncia un niño que si lo pronuncia algún otro. Obsérvese la siguiente comparación de la figura 7.1 (a) y (b):

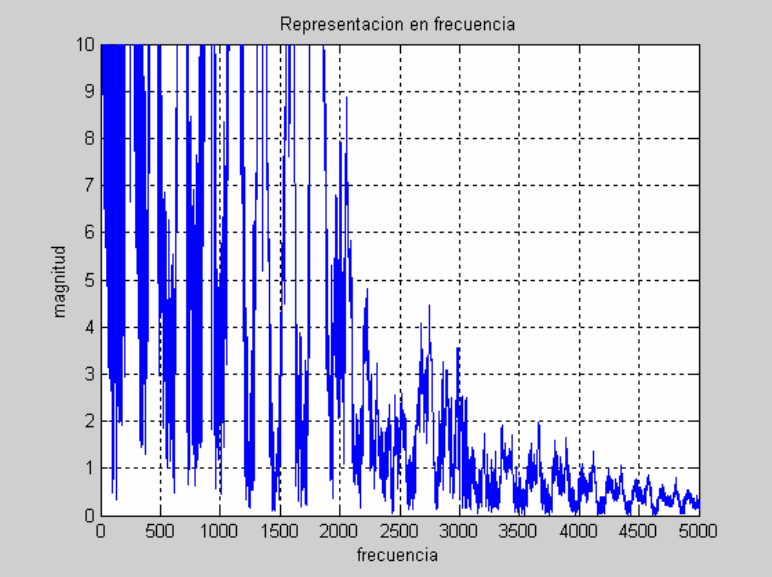


Figura 7.1 (a) Fonema “a” pronunciado por Aureliano

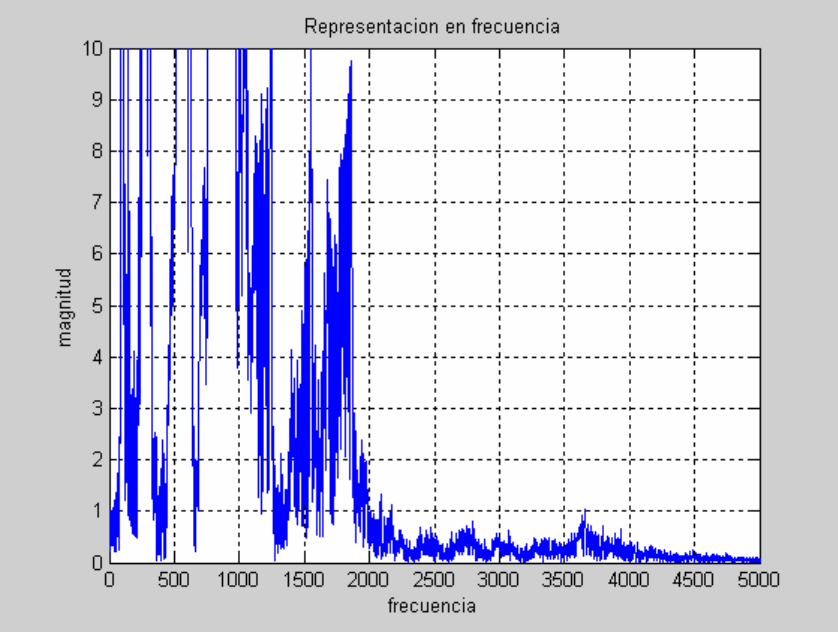


Figura 7.1 (b) Fonema “a” pronunciado por Lucy

Como se puede observar las representaciones en frecuencia aunque se trata del mismo fonema, son totalmente distintas, ya que son pronunciados por diferentes personas.

Entonces idealmente se esperaría que fuera fácil la identificación de la discrepancia entre dos fonemas, sin embargo, debido a su mermada función del habla se puede observar casi la misma representación en frecuencia para dos fonemas distintos pronunciados por la misma persona, por lo que se debe llevar a cabo un procedimiento que permita diferenciar los sonidos.

Para ello es necesario visualizar los sonidos como una señal y no como una serie de datos, como se menciona en [16]. El proceso antes descrito es precisamente, la base del presente sistema de reconocimiento, puesto que para la diferenciación de los sonidos es necesario llevar a cabo un enfoque gráfico al análisis de los sonidos.

7.1 Creación del sistema de reconocimiento de voz

Para poder diferenciar los sonidos que al oído pudieran parecer iguales, resulta menester buscar otra forma de visualizar las señales, por lo cual se optó por analizar la

magnitud del espectro de cada señal, pues, aunque su forma de onda sea similar, la escala de valores puede resultar distinta.

Como se muestra en la figura 7.1, la señal es sumamente irregular, luego entonces, encontrar un patrón en la misma podría resultar casi imposible. Para encontrar un patrón es posible someter la señal a un proceso de normalización en magnitud, para posteriormente realizar una transformación. Una vez que se obtiene la señal normalizada, ésta es almacenada en un vector de datos. Este vector se normaliza en magnitud con la ayuda de MATLAB® utilizando la siguiente expresión dada por:

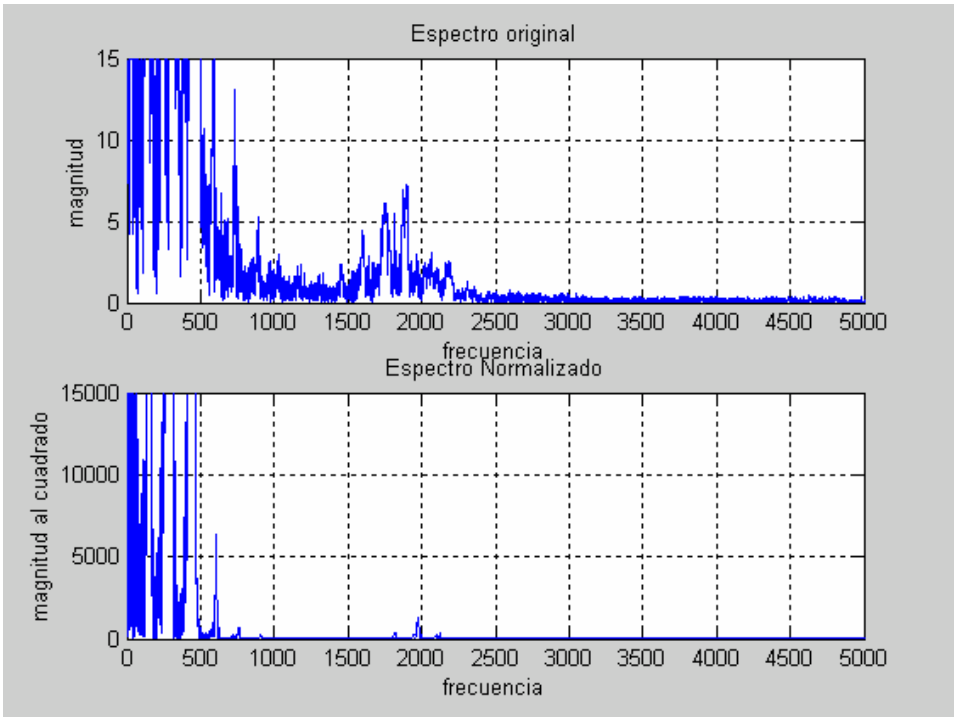
z=x./max(abs(x));

Entonces obtenemos un nuevo vector de datos *z* cuya magnitud se encuentra normalizada. A este nuevo vector de datos se le aplica el algoritmo de la Transformada Rápida de Fourier que posee MATLAB para obtener su representación en frecuencia. En la misma instrucción se obtiene la magnitud de este nuevo vector, así como también se eleva al cuadrado. Esto se lleva a cabo para poder escalar los máximos y mínimos más significativos de la señal, mientras que los menos significativos se atenúan. Cabe mencionar que esto también permite que la diferencia entre los máximos y mínimos de dos señales se incremente, pudiendo así minimizar la similitud entre las señales de dos sonidos distintos. La instrucción que lleva a cabo estos tres procesos o instrucciones se muestra a continuación y es parte del programa 1 que se puede consultar en el apéndice A.

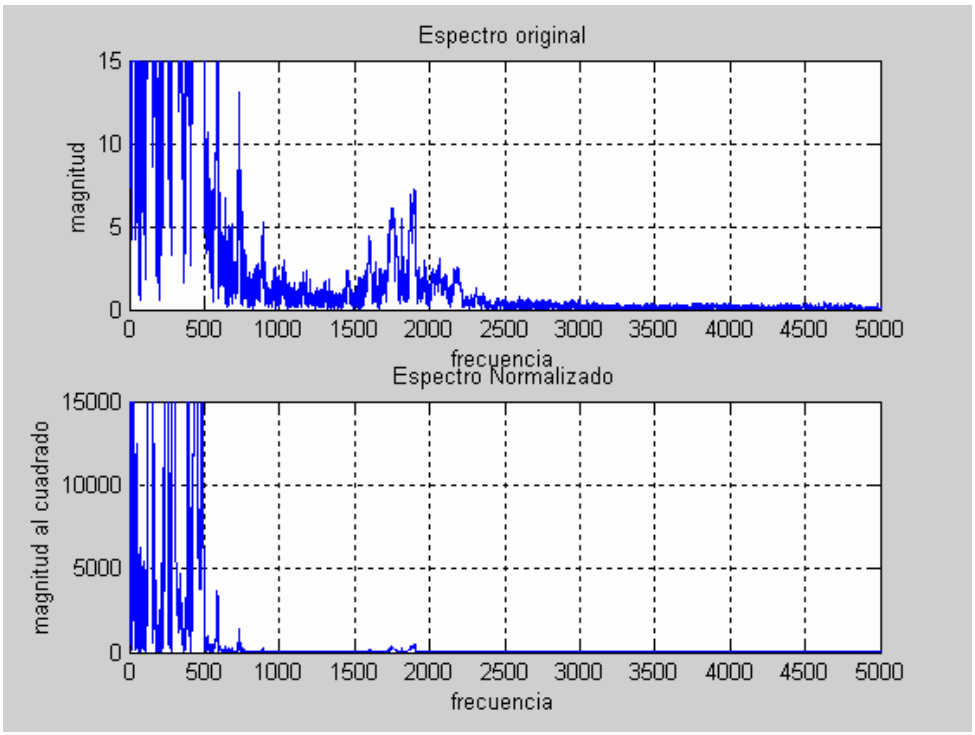
mag1=(abs(fft(z,size))).^2;

Para observar los beneficios de aplicar dicha instrucción se puede observar la figura 7.2, que corresponde a la pronunciación de los fonemas i y ke de Aureliano, que en su espectro original son muy similares, pero al aplicarles la normalización se pueden notar diferencias significativas.

El programa que se utilizó para obtener las siguientes gráficas se puede consultar en el Apéndice A bajo el nombre de *programa 2.*



(a)



(b)

Figura 7.2 Espectros original y normalizado para los fonemas “i”(a) y “ke”(b)

Este procedimiento no resulta suficiente para establecer diferencias significativas por lo que basándonos en el procedimiento propuesto en la fuente [16] se optó por separar las señales en distintas bandas de operación, pues los máximos y mínimos varían en distintos puntos de la señal. Se había optado primeramente en establecer una banda cada 400Hz, sin embargo esta separación no permitía una óptima diferenciación de los fonemas por lo que se redujeron las separaciones entre banda a 200Hz, lo cual mejora significativamente la resolución. El rango de frecuencias en el cual se lleva a cabo la separación es de 0 a 2500 Hz, pues es en este rango en que se encuentra la información significativa de las señales de voz. Para separar las bandas se utilizaron filtros pasabajas (por los rangos de frecuencia en que se encuentra la información que se desea analizar) Butterworth, utilizando la función *butter* de MATLAB en donde el usuario recibe los coeficientes del numerador y del denominador dándole al programa el valor del orden del filtro y de la frecuencia de corte, con lo cual obtenemos la función del filtro. La sintaxis del comando es como sigue:

[B,A]=butter(N,Wn);

donde B y A son los coeficientes del filtro, N el orden del mismo y Wn su frecuencia de corte.

Para aplicar dicho filtro se debe usa el comando *filter* con lo que se le introduce al filtro un vector de datos y se obtiene el vector resultante a la salida. El comando es el que se muestra a continuación:

y=filter(B,A,x)

donde *x* es la entrada, y *y* es la salida mientras que *B* y *A* son los coeficientes calculados mediante la función *butter*.

En la fuente [13] el lector puede encontrar información más detallada respecto al uso de los comandos arriba mencionados.

7.2 Bandas de operación

Las bandas de operación se establecieron cada 200 Hz y se separaron en bandas pares e impares con lo que podemos establecer diferentes zonas de trabajo para diferenciar los espectros. Los rangos de dichas bandas se muestran a continuación:

Para las bandas impares:

Banda	Rango de frecuencia
1i	100 – 300 Hz
2i	300 – 500 Hz
3i	500 – 700 Hz
4i	700 – 900 Hz
5i	900 – 1100 Hz
6i	1300 – 1500 Hz
7i	1500 – 1700 Hz
8i	1700 – 1900 Hz
9i	1900 – 2100 Hz
10i	2100 – 2300 Hz
11i	2300 – 2500 Hz
12i	2500 – 2700 Hz

Tabla 7.1 Bandas impares del sistema

Lo mismo para las bandas pares:

Banda	Rango de frecuencia
1p	0 – 200 Hz
2p	200 – 400 Hz
3p	400 – 600 Hz
4p	600 – 800 Hz
5p	800 – 1000 Hz
6p	1000 – 1200 Hz
7p	1200 – 1400 Hz
8p	1400 – 1600 Hz
9p	1600 – 1800 Hz
10p	1800 – 2000 Hz
11p	2000 – 2200 Hz
12p	2200 – 2400 Hz

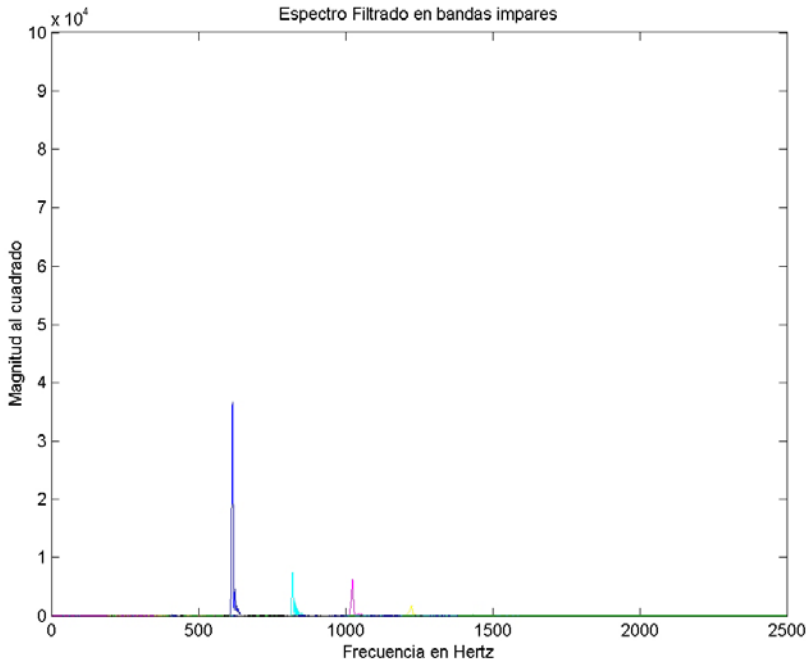
Tabla 7.2 Bandas pares del sistema

Una vez que se han establecido las diferentes bandas es posible diferenciar sonidos ya que presentan diferentes magnitudes en cada banda por lo que se puede obtener un algoritmo de reconocimiento para cada niño. El programa que permite

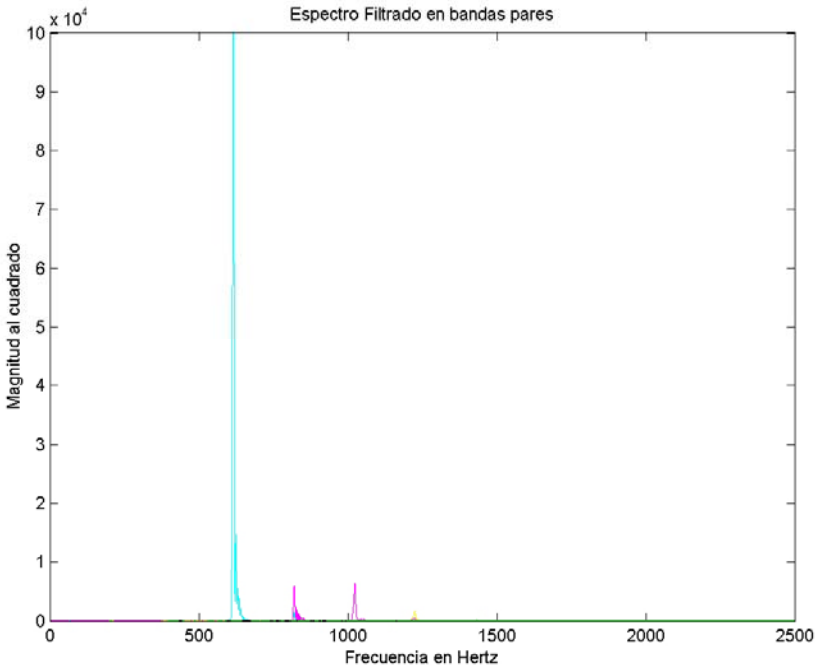
separar las señales en distintas bandas de operación y graficarlas se puede encontrar en el apéndice A bajo el nombre de Programa 2.

Una vez que se tienen las gráficas de todos los fonemas que se busca sean reconocidos se puede comenzar a buscar el patrón característico de cada niño. Para ello se deben comparar las gráficas de cada uno de los fonemas para así establecer constantes de comparación y buscar crear el algoritmo.

Las gráficas para los fonemas YO y DA pronunciados por Aureliano y separados en sus bandas pares e impares se muestran en las figuras 7.3 y 7.4.

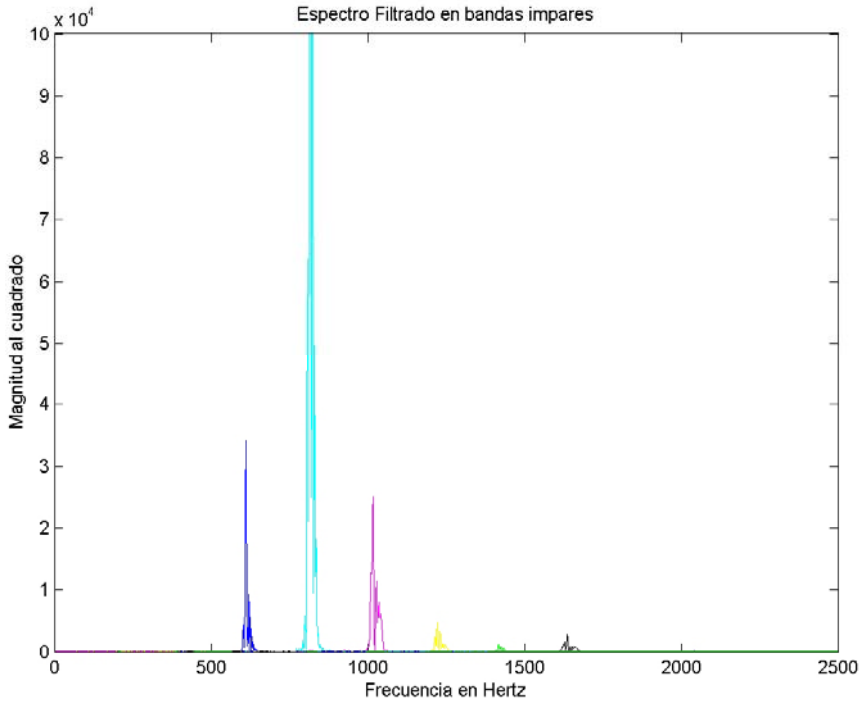


(a)

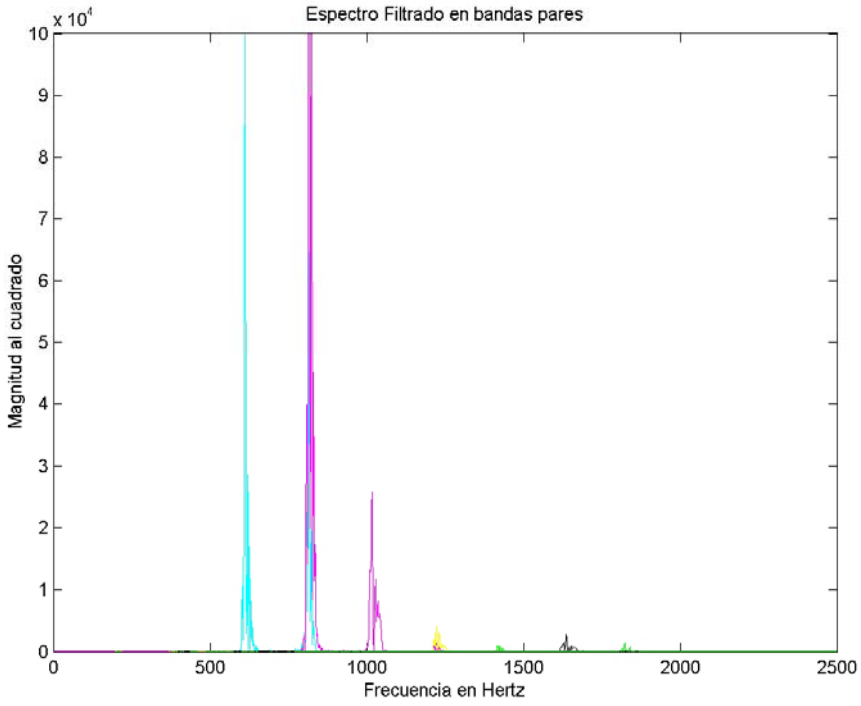


(b)

Figura 7.3 Fonema "Yo" pronunciado por Aureliano separado en sus bandas impares (a) y pares (b)



(a)



(b)

Figura 7.4 Fonema “Da” pronunciado por Aureliano separado en sus bandas impares (a) y pares (b)

Para diferenciar estos dos fonemas es posible declarar constantes en cada banda. Por ejemplo si se deseara diferenciar estos fonemas uno del otro declaramos una constante en la banda 4i (700 – 900 Hz) con un valor de 10000 puesto que el fonema “da” presenta un valor mucho mayor al fonema “yo” en dicha banda, luego entonces, si el máximo en la banda 4i es mayor que la constante declarada se elige el fonema “da”, si se da el caso contrario se opta por el fonema “yo”. En la figura 7.5 se muestra el algoritmo de reconocimiento creado para Aureliano

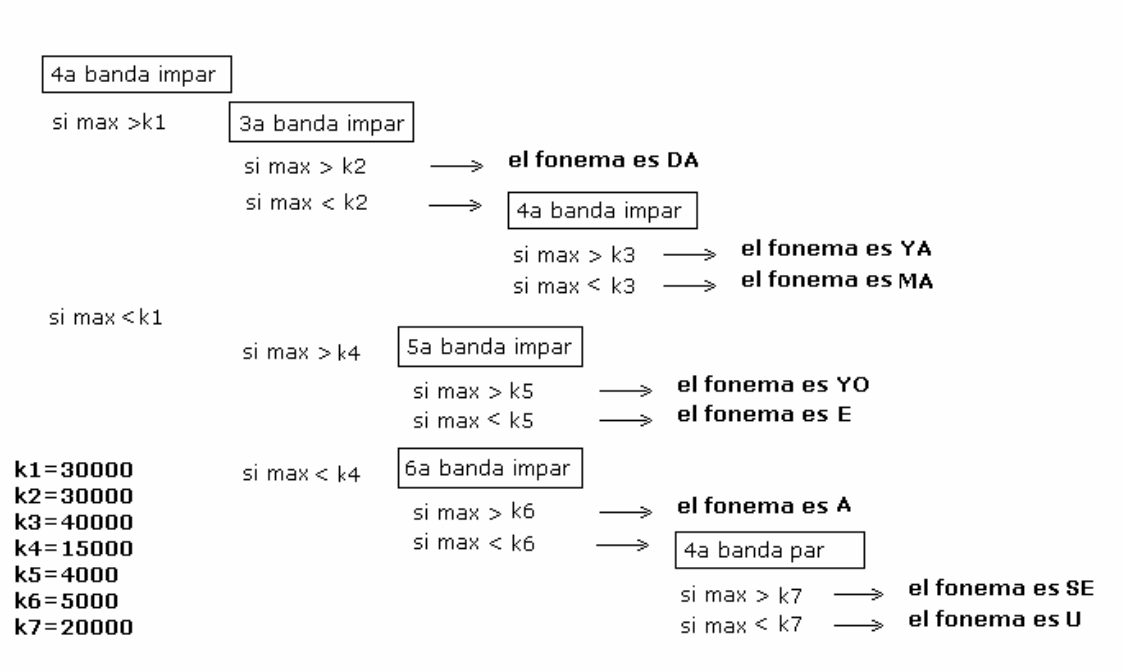


Figura 7.5 Algoritmo de reconocimiento para Aureliano

Este es el principio para crear los algoritmos de reconocimiento de cada uno de los niños. Los programas que permiten el reconocimiento de cada uno de los niños se pueden consultar en el Apéndice A con los nombres Programa Aureliano, Programa Lucy, Programa Octavio y Programa Israel.

De igual manera las imágenes correspondientes La totalidad de las imágenes separadas en bandas pares e impares, para cada niño y cada fonema pueden consultarse en el Disco compacto anexo al presente reporte.

7.3 Operación del Sistema

En el diseño del sistema se eligieron 8 fonemas para cada niño que son los que son considerados para reconocer. Los factores para elegir estos 8 fonemas distintos son su frecuencia de uso ya que forman palabras por sí solos, por ejemplo: yo, tu, ma, etc. Otro factor fue la gráfica de su espectro en frecuencia que presentaron a la salida, pues en algunos fonemas era imposible diferenciarlos y declarar las constantes de comparación.

7.3.1 Entrenamiento y creación del sistema

El sistema fue creado utilizando las muestras de voz capturadas en la primera fase del proyecto para obtener así los fonemas que fueron más reconocidos.

A continuación se muestra una lista con los fonemas elegidos para cada niño:

NIÑO	FONEMAS ELEGIDOS
Aureliano	a, e, da, ma, yo, se, ya, u
Lucy	o, tu, si, ya, yo, a, e, i
Daniela	a, e, yo, ka, ma, u , tu, ya
Octavio	i, u, o, ya, a, de, ba, e
Israel	A, e, i , ka, ke, ma, me, o

Tabla 7.3 Lista de fonemas integrados al sistema

Para cualquier fonema no considerado en la tabla 7.3 no se podrá llevar a cabo el reconocimiento, pues habría que dar de alta el fonema en el algoritmo incrementando así la complejidad del mismo. Por la similitud en su espectro en frecuencia, quizá si se le introdujera el fonema “na” al programa de Aureliano desplegaría “ma”, lo cual es un resultado irreal, sin embargo cambiando el algoritmo se podría incorporar al sistema.

7.4 Interfaz Gráfica (GUI)

Con el objeto de facilitar la interacción del usuario con el programa antes descrito se creó una interfaz gráfica utilizando la herramienta GUIDE incorporada en MATLAB. Para crear una interfaz gráfica únicamente es necesario teclear el comando *guide* en la ventana de comando y se abrirá una ventana donde se le permite al usuario arrastrar herramientas para crear la interfaz deseada.

GUIDE, el ambiente de MATLAB® para la creación de interfases gráficas de usuario (GUI) provee un conjunto de herramientas para crear este tipo de interfases.

Estas herramientas simplifican significativamente el proceso de diseño y construcción de las GUIs. Se pueden utilizar las herramientas de GUIDE para

- Esquematizar la GUI. Es posible crear un bosquejo de una GUI fácilmente, arrastrando al área de trabajo los componentes de la misma, tales como paneles, botones, campos de texto, barras deslizadoras, menús, etc.
- Programar la GUI. GUIDE genera automáticamente un archivo M que controla como opera la GUI. El archivo M inicializa la GUI y contiene el funcionamiento de todos los códigos de cada componente. Usando el editor de archivos M, se puede agregar código las callbacks (código que indica el funcionamiento) para que lleven a cabo las funciones que el usuario espera de cada componente.

La interfaz creada en este proyecto, permite seleccionar al niño del cual se desea llevar a cabo el reconocimiento de Voz. La figura 7.6 muestra la ventana principal de la interfaz creada.

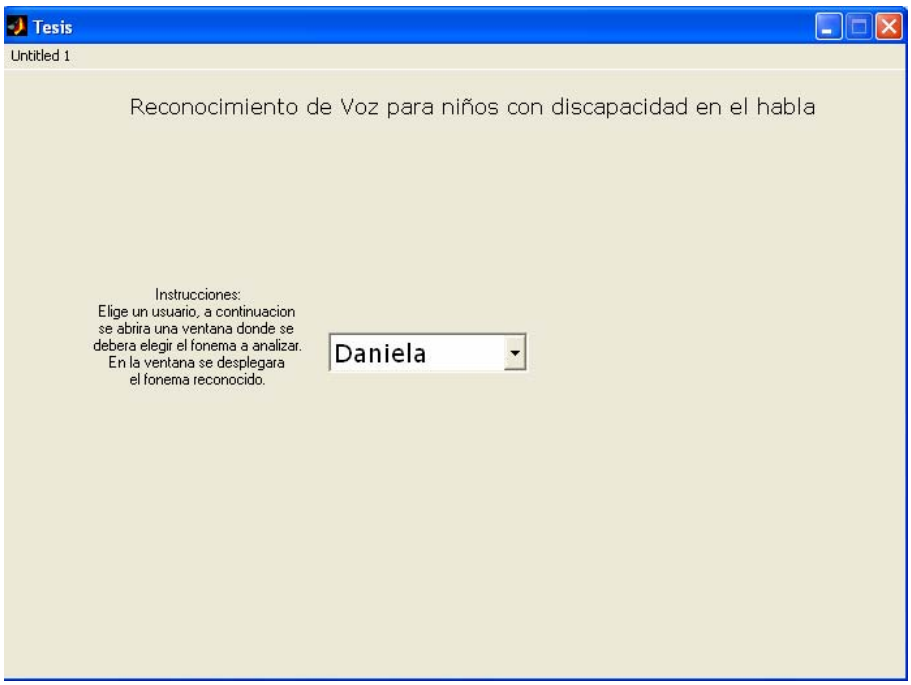


Figura 7.6 Interfaz gráfica del sistema de reconocimiento de Voz.

Una vez elegido el niño se abre una nueva ventana en la cual se puede elegir de 8 diferentes fonemas, cada uno asociado a uno de los *wavs* obtenidos en la etapa de muestras de voz. A este archivo elegido se le aplica el algoritmo y en el cuadro de texto se despliega el fonema que se obtuvo. En la figura 7.7 se observa la interfaz para Octavio.

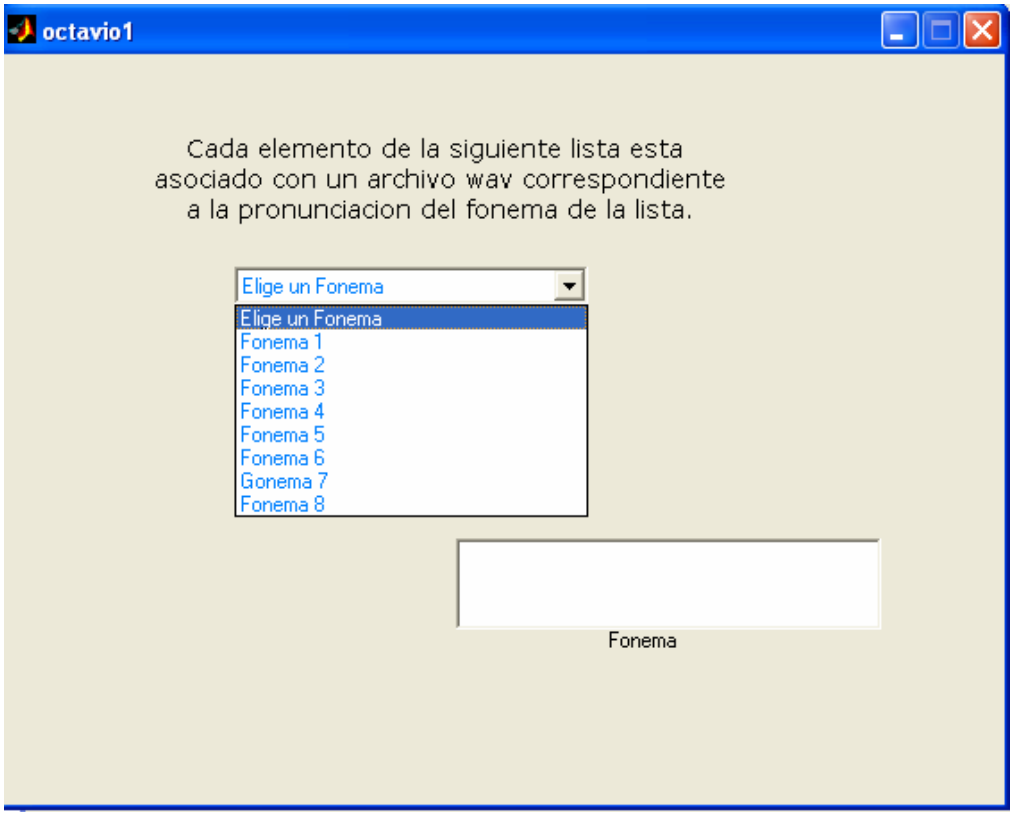


Figura 7.7 Interfaz para Octavio

Al elegir un fonema de la lista, que como se ha mencionado, están asociados con los *wavs* obtenidos en la recolección de muestras, se despliega el fonema reconocido. Obsérvese la figura 7.8

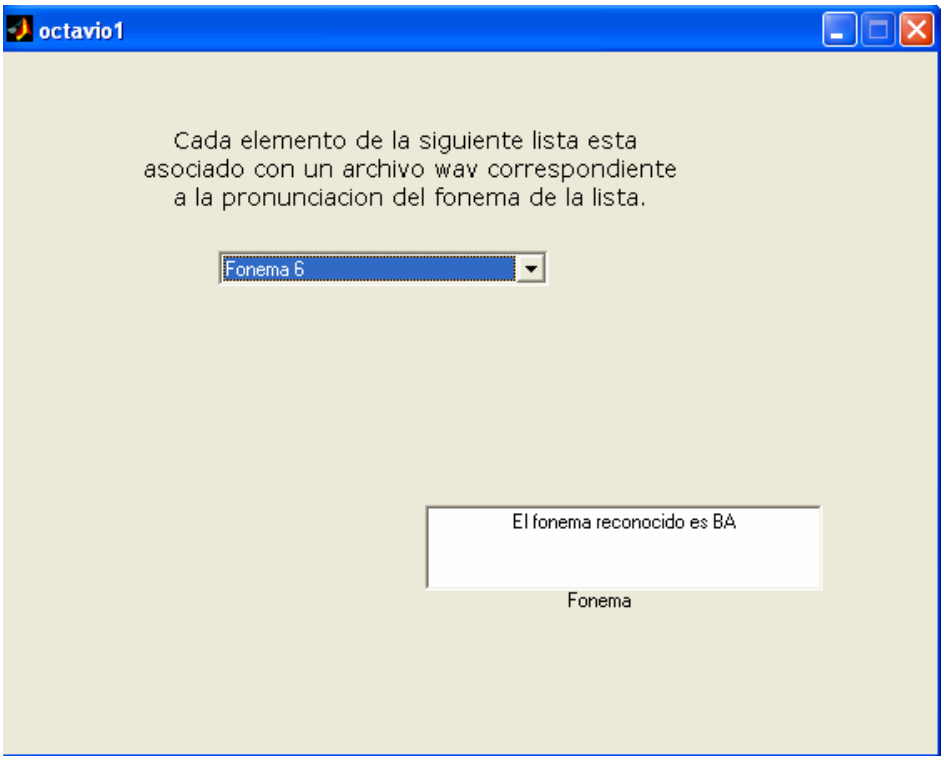


Figura 7.8 Fonema reconocido

Esta interfaz no es factible para usarse con el objeto de llevar a cabo pruebas, pues los *wavs* que utiliza son los que se consideraron para crear el algoritmo de reconocimiento, por lo que fue necesario la creación de una interfaz adicional, así como también la obtención de nuevos archivos *wav* que permitieran verificar un desempeño satisfactorio del sistema de reconocimiento de voz. Dichas pruebas se muestran en el siguiente capítulo. El código de la interfase se puede consultar en el Apéndice B.